

Examination Computational Statistics

Linköpings Universitet, IDA, Statistik

Course code and name:	732A90 Computational Statistics
Date:	2019/03/19, 8–13
Assisting teacher:	Krzysztof Bartoszek
Allowed aids:	Printed books, 100 page computer document, and material in the zip file extra_material.zip
Grades:	A= [18 – 20] points B= [16 – 18) points C= [14 – 16) points D= [12 – 14) points E= [10 – 12) points F= [0 – 10) points
Instructions:	Provide a detailed report that includes plots, conclusions and interpretations. If you are unable to include a plot in your solution file clearly indicate the section of R code that generates it. Give motivated answers to the questions. If an answer is not motivated, the points are reduced. Provide all necessary codes in an appendix. In a number of questions you are asked to do plots. Make sure that they are informative, have correctly labelled axes, informative axes limits and are correctly described. Points may be deducted for poorly done graphs. Name your solution files as: [your anonymous exam account]_[own file description].[format] There are TWO assignments (with sub-questions) to solve. Provide a separate solution file for each assignment. Include all R code that was used to obtain your answers in your solution files. Make sure it is clear which code section corresponds to which question.

NOTE: If you fail to do a part on which subsequent question(s) depend on describe (maybe using dummy data, partial code e.t.c.) how you would do them given you had done that part. You *might* be eligible for partial points.

Assignment 1 (10p)

Trigonometric functions are crucial in statistical and mathematical operations, hence good numerical implementations of them are very important. Your task is to implement functions to calculate the sine, cosine and tangent functions and explore how numerically stable they are. The Taylor expansions of these functions around 0 are

$$\sin(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1}$$

$$\cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n}$$

$$\tan(x) = \sum_{n=0}^{\infty} \frac{(-1)^{n-1} 2^{2n} (2^{2n} - 1) B_{2n}}{(2n)!} x^{2n-1},$$

where B_k are the so-called Bernoulli numbers which can be obtained as `pracma::bernoulli(k,0)`. Furthermore, we know that

$$\tan(x) = \frac{\sin(x)}{\cos(x)}.$$

For each of these trigonometric functions implement a function that approximates them, through their Taylor expansions. Your functions should take as a parameter N —the desired degree of the expansion and of course x . Study how your implementations behave for different values of N and values of x . You may use R's `sin()`, `cos()` and `tan()` functions as if they return the true values. For the $\tan(x)$ function also implement a procedure that calculates it using the quotient of the sine and cosine functions. Compare how this behaves. Explain what you observe, present plots and/or tables documenting the behaviour and comparing with the correct values.

TIP: the relative error (or variations around it) $|\hat{f} - f_{\text{true}}|/f_{\text{true}}$ is sometimes a good way to assess the approximation to a function.

Assignment 2 (10p)

In the file `data.RData` you will find a matrix with 100 independent observations coming from a 10-dimensional normal distribution. The generating distribution has mean $\vec{0} \in \mathbb{R}^{10}$ and a very special covariance structure $\Sigma \in \mathbb{R}^{10 \times 10}$. Namely Σ has 1s on its diagonal, and either 0 or 0.1 outside of its diagonal. Your aim is to implement a genetic algorithm to find which entries are non-zero on the off-diagonals. If it is helpful you may assume that at most 10% of the off-diagonal entries are non-zero.

Question 2.1 (2p)

An initial step when implementing a genetic algorithm is the decision on how individuals in the population will be represented. Decide yourself on the representation and implement a function that transforms an individual from the population into the Σ matrix.

TIP: you might find the following code useful

```
S[lower.tri(S,diag=FALSE)]<-t(S)[lower.tri(t(S),diag=FALSE)].
```

TIP: you might find the functions

`matrixcalc::is.positive.definite()` and `matrixcalc::is.symmetric.matrix()` useful.

Question 2.2 (3p)

The next step is to implement a selection procedure in order to choose the pool of individuals that will be allowed to contribute to the next generation. Implement a function that chooses individuals for reproduction based on their associated likelihood values.

TIP: you may want to consider the function `mvtnorm::dmvnorm()`.

Question 2.3 (5p)

Implement all the other necessary components of the genetic algorithm and using it find the maximum likelihood estimator of Σ 's structure. Take a population size of about 100. It is recommended that you first try your code on a smaller population (say 6) and only after it runs without error, run it on the population of size 100. Follow your population for 100 generations. Provide example calls to your code. If your code runs too long decrease the population size and number of generations. If the running time is fine, but you do not observe good estimations increase both. Design and implement an informed method of generating the initial population. Use the fact that you know that the population has mean $\vec{0}$.

Visualize the population's behaviour with the number of generations for mutation probability 0.0077 and 0.5 and provide comments. At which generation was the best value found? Can you explain the observed behaviour, especially when taking into account the mutation probability?